Scalable and Efficient Big Data Analytics: The LeanBigData Approach

Ricardo Jimenez[1], Marta Patino[2], Valerio Vianello[2], Ivan Brondino[1], Ricardo Vilaca[1], Jorge Teixeira[3], Miguel Biscaia[3], Giannis Drossis[4], Damien Michel[4], Chryssi Birliraki[4], George Margetis[4], Antonis Argyros[4], Constantine Stephanidis[4], Luigi Sgaglione[5], Gaetano Papale[5], Giavanni Mazzeo[5], Ferdinando Campanile[6], Marc Sole[7], Victor Muntés-Mulero[7], David Solans[7], Alberto Huelamo[7], Pavlos Kranas[8], Dora Varvarigou[8], Vrettos Moulos[8] and Fotis Aisopos[8]

[1]LeanXcale, Madrid, Spain
[2]Universidad Politecnica de Madrid, Madrid, Spain
[3]Altice Labs, Aveiro, Portugal
[4]Institute of Computer Science, Foundation for Research and Technology Hellas & Computer Science Department, University of Crete, Heraklion, Greece
[5]University of Naples “Parthenope”, Naples, Italy
[6]Sync Lab srlSync Lab srl, Naples, Italy
[7]CA Technologies, Barcelona, Spain
[8]National Technical University of Athens & ICCS, Athens, Greece

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Abstract. One of the major problems in enterprise data management lies in the separation of databases between operational databases and data warehouses. This separation is motivated by the different capabilities of OLTP and OLAP data management systems. Due to this separation copies from the operational databases to the data warehouses should be performed periodically. These copies are performed by a process call Extract-Transform-Load (ETL) that turns out to amount to 80% of the budget of performing business analytics. LeanBigData main goal has been to address this major pain by providing a real-time big data platform providing both functions, OLTP and OLAP, in a single data management solution. The way to achieve this goal has been to leverage an ultra-scalable OLTP database, LeanXcale, and develop a new OLAP engine that works directly over the operational data. The platform is based on a novel storage engine that provides extreme levels of efficiency. The platform has also an integrated parallel-distributed CEP that scales the processing of streaming data and that can be combined with the processing of data at rest at the new OLTP+OLAP database to address a wide variety of data management problems. LeanBigData has a bigger vision and aims at providing and end-to-end analytics platform. This platform provides a visual workbench that enables data scientist to perform discovery of new insights. The platform is also enriched with a subsystem that performs anomaly detection and root cause analysis that works with the new developed system and enables to perform this analysis over streaming data. The LeanBigData platform has been validated by four real-world use case scenarios cloud data centre monitoring, fraud detection in direct debit operations, sentiment analysis in social networks and targeted advertisement.

Keywords. Big Data, Real-Time Big Data, SQL, OLTP, OLAP.

Rahim R., Rahim R. and Rahim R.
Scalable and Efficient Big Data Analytics - The LeanBigData Approach.
DOI: 10.5220/0007903100920111
In *European Space project on Smart Systems, Big Data, Future Internet - Towards Serving the Grand Societal Challenges (EPS Rome 2016 2016)*, pages 92-111
ISBN: 978-989-758-207-3
Copyright © 2016 by SCITEPRESS – Science and Technology Publications, Lda. All rights reserved

1 Introduction - Real-Time Big Data Analytics

The current enterprise world uses two main technologies for data management, operational databases and data warehouses. Operational databases provide consistency guarantees for data that is being updated frequently. These consistency guarantees are known as ACID properties or transactional guarantees. Operational databases are On Line Transactional Processing (OLTP) systems able to process short transactions with a high fraction of update transactions. Data warehouses are used for performing business analytics. They are able to answer large analytical queries in an online manner. They are On Line Analytical Processing (OLAP) systems.

Unfortunately, both systems are bad at processing the other workload. So enterprises need to keep both kind of databases. This results in the need to copy periodically the data from the operational databases into the data warehouse in process, so called, Extract-Transform-Load (ETL). This process is estimated to be an 80% of the budget of doing business analytics a total non-sense. LeanBigData had as main goal to solve this pain in enterprise data management by conceiving a new data management system that is able to solve both kind of workloads, thus avoiding the need for ETLs. The result is a new database, LeanXcale, with both OLTP and OLAP capabilities scaling in both dimensions of the workload.

LeanBigData has also produced a set of technologies to accompany the real-time big data platform. First, it provides support for doing visual analytics directly on top of the new data management system by means of a new visual analytics workbench. Second, it has developed a scalable technology for data stream processing that is integrated with the database, enabling to combine the processing of both streaming data and data at rest, enabling to correlate streaming data with data stored in the operational database and/or update the operational database. Third, it has developed a configurable data collection system to get data from different sources from sensors to other sources of data.

SCIENCE AND TECHNOLOGY PUBLICATIONS

2 The LeanBigData Platform

The starting point from the LeanBigData platform was the LeanXcale original version. It was an ultra-scalable OLTP database based on an open source stack for the storage and persistence layers. One of the main goals was to build a distributed query engine that would leverage the ultra-scalability of the transactional database to answer analytical queries reading the data directly from the operational database resulting in a database blending the two capabilities, OLTP and OLAP. The original LeanXcale at the start of the project was based on an open source stack based including HBase, a key-value data store, as a storage engine, and HDFS, a distributed file system, as persistence layer. This was a good starting point, but the CPU cost per row with this open source stack was too high for an OLAP engine. Although a scalable OLAP engine could be built, it would require a high number of nodes, being not a competitive technology with data warehouses. For this purpose, within LeanBigData, one of the core objectives was to build an ultra-efficient key-value data store, KiVi, that would serve as new storage engine. Another major improvement for LeanXcale has been to improve the efficiency of the transactional processing as much as possible to reduce the overhead introduced by it. The goal has been to leverage the asynchronous nature of the underlying transactional processing algorithm to reduce the cost of messaging inherent to distribution and introduce batching at all levels of message exchange.

2.1 Ultra-Efficient Key-value Data Store

Existing storage technologies for operational data management basically fall in two categories. Storage technologies coming from relational databases and those coming from the new key-value data stores. The former are good at range queries. The latter are good at random updates. However, they are bad at the other workload. In the context of LeanBigData, a new key-value data store has been conceived, KiVi, that is based on a new data structure combining the capabilities of both relational databases and key-value data stores storage engines, being good at both kind of workloads range queries and random updates. Another major advancement in KiVi has been that it has been designed from the beginning to work efficiently in multi-core NUMA architectures. Current architectures have multiple CPU sockets and each CPU multiple cores. This is resulting in architectures in which the access to memory is not uniform, due to the access to the memory close to the core is faster than the access to memory far from the core (e.g. on a different socket), resulting in the so-called Non-Uniform Memory Access or NUMA architectures. KiVi also brings a very novel feature that lies in its ability to be reconfigured dynamically without stopping the ongoing processing. This feature is being exploited to provide elasticity and dynamic load balancing the storage engine.

2.2 Ultra-Efficient Transactional Processing

The starting point for the transactional processing was an ultra-scalable transactional system. However, its distributed nature resulted in a non-neglectable cost due to messaging across nodes for actions related to data items and transactions. It was processing all messages with a traditional reactive strategy, in which each update data item resulted in a message with the associated CPU cost for sending and receiving the message at the two ends, in addition to the network latency, and each action related to a transaction resulted as well in individual messages per transaction and the associated increase in latency. The underlying algorithms and protocols for attaining the scalability of the transactional processing in LeanXcale are asynchronous in nature. This feature has been exploited to change all the message processing to a proactive strategy. Basically, a period is established that is used by all components related to transactional processing. During this period all actions related to data items or transactions that should be sent to a particular component are buffered in a message. After the period is exhausted (or earlier, if the buffer gets full before), the message is sent to the target component. A typical period length can be 10 milliseconds that results in batching in a single message the interaction for thousands of data items or transactions. Thus, the fix cost of sending a message is amortized across 1000s of items, making it neglectable. This is an important optimization in a transactional system that aims at processing millions of update transactions per second resulting in many messages under the former reactive strategy. Additionally, the average CPU cost per transaction with the new reactive strategy has been highly reduced, making the cost of adding transactions very low and very affordable.

2.3 Distributed SQL Query Engine

As aforementioned one of the major goals of the project was to develop a distributed query engine to provide OLAP capabilities to LeanXcale that would be combined with its already existing OLTP capabilities. In LeanBigData a new OLAP engine has been built. The OLAP engine is able to run a query plan across multiple nodes. More concretely, it introduces intra-query and intra-operator parallelism that enables to lower the time to process a query. At each node, a query engine is run. Each query engine instance processes a fraction of the query. One of the query engine instances is contacted by an application and sends the query in SQL. The query engine compiles the SQL and generates a query plan for the query. This query plan is centralized. The new OLAP engine takes a centralized query plan and modifies it as follows. At the bottom there are scan operators that access directly the data. These scan operators are projected over the local regions on the nodes where each of the query engine instances run. Then, for each stateful operator in the plan, a reshuffling operator is introduced that redistributes data across instances. This operator guarantees that each query engine instance receives all the data needed for a given operator to produce coherent results. For instance, if the operator is an aggregate being run by a set of query instances, the aggregate operator at each instance will receive a set of values to be aggregated together so it can produce the correct result (e.g. all the values corresponding to a given country, that is, values from the same country go to a particular instance and do not get split across several instances what would produce incorrect results). Finally, at the root of the query plan the query engine instance that initially generated the query collects all the results from all other query engines to produce the final result set and send it to the client application.

2.4 Visual Analytics Workbench

One of the goals in business analytics is the discovery of new insights. Discovery of new insights requires the ability to perform ad-hoc analytical queries and transform the results in order to discover relations across data. In order to support this activity, in LeanBigData a visual analytics workbench has been produced. This visual workbench enables to issue analytical queries into the LeanXcale real-time analytics database. Their result sets can then be massaged and transformed via algebraic operators generating new results than can be transformed, visualized and analyzed iteratively till new insights are unveiled. The algebraic operators enable to perform different tasks such as selections, projections, aggregations, etc. The visualization operators enable to visualize with different kind of charts any of the partial results such as line charts, pie charts, etc. The resulting network of algebraic and visualization operators form a workflow that basically performs visual analytics. The workbench is based on a drag and drop interface that enables an intuitive interaction with the user. The implementation of the workbench is also interesting. It basically stores partial results as temporary tables in the LeanXcale database. The algebraic operators are translated into SQL queries that are executed over the temporary tables using the OLAP engine, and therefore getting fast responses exploiting the underlying parallelism.

2.5 Distributed Complex Event Processing System

Complex Event Processing (CEP) is a novel paradigm for analyzing in real-time data captured from heterogeneous data sources. Instead of storing the data and then process it, the data is processed on the fly, as soon as it is received, or at most a window of data is stored in memory. CEP queries are continuous queries run on a stream of events. Continuous queries are modeled as graphs where nodes are CEP operators and arrows are stream of events. CEP operators are computational boxes that process events received over the incoming stream and produce output events on the outgoing streams. CEP operators can be either stateless or stateful, depending on whether they operate on the current event (tuple) or on a set of events (window). In the last few years several implementations went out to the consumer market from both academy (such as Borealis [1]) and industry (such as Infosphere [2] and Esper [1]). Later on, solutions like Storm [2] and S4 [3] followed a similar approach to the one of StreamCloud [3] for addressing the scalability of CEP systems in order to be able to process in real-time the increasing amount of data being produced every day. With these systems, a CEP can run a continuous query in a distributed and parallel way over several machines, which in turn increases the system throughput in terms of number of tuples processed per second. LeanBigData CEP (LBD-CEP) adds efficiency to this parallel-distributed processing being able to reach higher throughput using less resources. LBD-CEP improves the network management (in current distributed CEPs about 50% of CPU is used for networking), reduces the inefficiency of the garbage collection by implementing techniques such as object reutilization and takes advantage of the novel Non Uniform Memory Access (NUMA) multicore architectures by minimizing the time spent in context switching of CEP threads/processes.

SCIENCE AND TECHNOLOGY PUBLICATIONS

The main components of the LBD-CEP are depicted in Figure 1. The data comes from sensors or other sources that produce a continuous stream of data. The Data Collection Framework (DCM) is a middleware used to collect data from the sensors and convert these heterogeneous data into a common format. Furthermore the DCM preprocesses on the edge in order to clean the data for further processing by the LBD-CEP. The CEP has a distributed architecture with three main components: the *Orchestrator*, *Instance Managers* and the *Reliable Registry*. Instance Managers are the processing components. They are single threaded CEP processes. The Orchestrator is the component in charge of distributing queries/sub-queries among the Instance Managers available in the cluster and monitoring the cluster for failures and performance bottlenecks. The Reliable Registry stores relevant information regarding the deployment of the instance managers and queries.

Client applications can interact with the LBD-CEP using a JCEPC driver. The JCEPC driver hides from the applications the complexity of the CEP cluster and presents them the cluster as a black box that applications can use to run parallel distributed

[1] http://www.espertech.com/esper
[2] http://storm.apache.org
[3] http://incubator.apache.org/s4

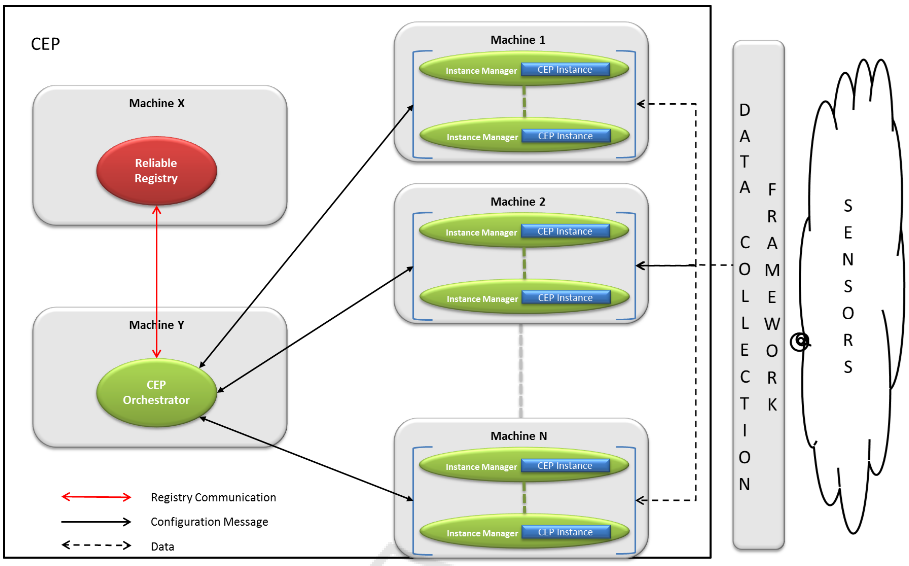

Fig. 1. Distributed complex event processing system.

continuous queries. Applications can create and deploy continuous queries using the JCEPC driver as well as register to the source streams and subscribe to output streams of these queries. During the deployment the JCEPC driver takes care of splitting a query into sub-queries and deploys them in the CEP cluster. Some of those sub-queries can be parallelized if needed. Registering/subscribing to source/output streams of a query is not trivial when the continuous query is parallelized because the stream itself is spread among all the instances of the query/sub-queries. Using the JCEPC driver, applications do not face the problem of discovering a stream location and managing the stream parallelization because the JCEPC driver hides also this complexity and present to the application a simple local stream to be fed.

LBD-CEP has been developed focusing on three main properties: scalability, flexibility and usability:

– Scalability: it is the most demanding feature for a CEP engine. LBD-CEP is able to scale with the number of queries using the inter-query parallelism. Continuous queries are deployed on several different nodes of the CEP cluster. LBD-CEP is also able to scale with the complexity of queries. This is achieved with the intra-query parallelism where a query is divided into sub-queries and then parallelized. Finally using the two previous techniques LBD-CEP is also able to scale with the stream volume.
– Flexibility: LBD-CEP is able to process streaming events and correlate them with data kept in external data stores. It also allows easily developing new custom operators and plugging them into the engine
– Usability: LBD-CEP has a user friendly interface for both query and data manipulation. The installation and configuration of the engine is also straightforward.

2.6 Anomaly Detection

Anomaly detection is a common need in many scenarios that involve the usage of an OLTP engine. Since the output of anomaly detection is typically needed as soon as possible to reduce the reaction time, the anomaly detection in the platform has been implemented as part of the Complex Event Processing subsystem, which processes incoming streams of data. We have also implemented Root-cause Analysis techniques, although we do not provide details in this chapter because of space limitations.

Although anomaly detection is a common need, sometimes the anomaly detectors required for a particular task are very specific. In some other cases generic approaches might work well enough in practice. In the LeanBigData project examples of both strategies have been developed. First, a generic stream clustering technique has been developed. Second, for each use case, several custom anomaly detectors have been created.

Performing anomaly detection in evolving data is difficult for a couple of reasons: an apparent anomaly at one time instance may not be an anomaly as it could be just a change of state in a system. There are basically three ways in which an anomaly occurring in a time series may be defined: (i) an event within a time series may be anomalous; (ii) a sub-sequence within a time series may be anomalous; or (iii) an entire time series may be anomalous with respect to previous time series. Different techniques and methods are required to detect different types of value, structural and contextual anomalies, which should be adherent to following challenges:

SCIENCE AND TECHNOLOGY PUBLICATIONS

- *Definition.* Defining a "normal" region that encompasses every possible normal behaviour is very difficult. The boundary between normal and anomalous behaviour is often not precise.
- *Masking.* A single anomaly is easier to detect than a series of anomalous readouts. The density of anomalies may render observations "normal", thereby making the task of defining normal behaviour more difficult.
- *Stationary.* In many domains normal behaviour keeps evolving and a current notion of normal behaviour might not be sufficiently representative in the future. What is now abnormal can be normal in the future.
- *Scaling.* The exact notion of an anomaly is different for different application domains. Thus applying a technique developed in one domain to another, is not straightforward.
- *Labels.* Availability of labeled data for training/validation of models used by anomaly detection techniques is usually a major issue.
- *Data Variations.* Often the data contains noise that tends to be similar to the actual anomalies and hence is difficult to distinguish and remove.
- *Infinite Size and High Speed.* Data in a data center tend to be of infinite size and continuously flowing at very high speed. Due to this complexity, it is impossible to store data and processing off-line is computationally very expensive.
- *Dynamic Nature.* Data behaviour keeps on changing over time. An anomaly detection technique developed on partial data must be updated with new incoming data.

- *Lack of Global View of Data.* In the traditional anomaly detection problem, global view of the data is available, however, in data centers, the global view of data is not available.
- *High Dimensional Data.* High dimensionality further magnifies the challenge of working with data due to the curse of dimensionality. In such spaces, all pairs of points tend to look almost equidistant from one another.
- *Nature of Input Data.* Input is generally a collection of data instances, which can be: object, record, point, vector, pattern, event, case, sample, observation, or entity. The attributes can be of different types such as: binary, categorical, or continuous. Each data instance can consist of: one attribute (univariate), or multiple attributes (multivariate).
- *Output of Anomaly.* Typically, the outputs produced by anomaly detection techniques are one of the two types: Scores or Labels.

Due to evolving nature of the data, unsupervised methods are recommended. Big data summarization requires lesser storage and extremely shorter time to get processed and retrieved. Stream clustering is an efficient strategy against mining of evolving big data and clustering based anomaly detection is an effective way to deal with all the big data and anomaly detection challenges. We evaluated big data stream clustering algorithms [4] and found DenStream [5] better than the other stream clustering algorithms [6]. DenStream works better because (a) it is incremental in nature, (b) it scans data single time, (c) it handles concept evolution, (d) it is robust with outliers and (e) it is non-parametric.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

2.7 Visualization

An important aspect regarding big data processing and analysis is data visualization and user interaction. The main questions that should be addressed towards this dimension is how to handle the visualization of the results of big data analysis and how a user can interact with it in a natural way, always in a context–centric approach. In the LeanBigData project, a Data Centre 3D Visualization application has been developed [7], showcasing large-scale data centers infrastructure monitoring and management. The main objective of this application in the scope of the project is to provide intuitive and rich gestural interaction with big data 3D visualization.

The Data Center 3D Visualization application aims at assisting data center experts to get an overview of the state of a specific data center room. Additionally, the application facilitates the inspection of the racks and servers and warns the users about situations such as anomalies regarding a particular set of servers, which may result in malfunctions or degraded operation and, as such, need further investigation. Taking into account the various potential contexts of use, ranging from an office to a control room, the application can be deployed in a laptop or a PC, featuring touch and mouse interaction with desktop setups, as well as gestures-based interaction on large displays in a manner similar to [8] and [9]

The main screen of the application comprises a virtual representation of a data center room and the basic interactive UI components (e.g., navigation buttons), as shown in

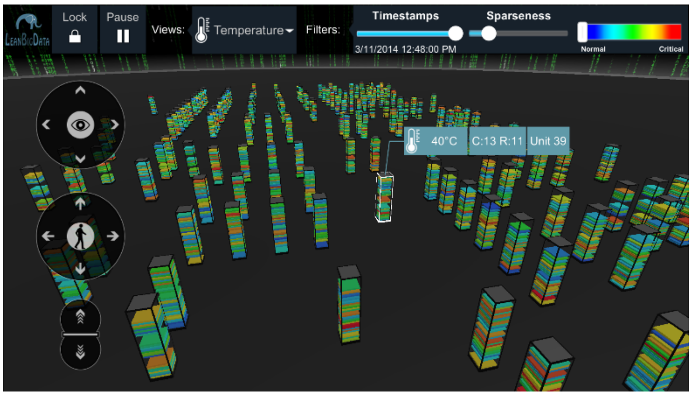

Fig. 2. Main Screen view.

Fig. 3. Gesture-based interaction.

SCITEPRESS
SCIENCE AND TECHNOLOGY PUBLICATIONS

Figure 2 and 3. All the servers of the room are grouped and displayed as 3D racks, arranged as a grid in 3D space that matches their physical location. Each rack may contain at most 40 units, each displayed as a slice with a specific color that signifies its current status regarding temperature, CPU load, power consumption, network Bytes In/Out and Disk I/O. In particular, a color-coded scale ranging from blue to red has been specified indicating the criticality level of each server with respect to any of the aforementioned parameters. Upon the selection of one of the servers' properties from the main menu, the displayed data visualization changes accordingly, displaying the criticality level of each server for the selected property. Thus, users can instantly switch among different server properties, perceive immediately the big picture regarding the relevant state of the Data Center and take appropriate actions, if needed. The virtual environment that encloses the scene is spherical and the servers' grid is placed in the center, so that users can have a 360 degrees overview.

In addition, the system offers filtering options, such as regulation of the sparseness of the servers in space as well as selection of the servers' status criticality level, allowing users to manipulate the information provided and focus on specific aspects of the data center.

Navigation and Interaction Design. Navigation in the scene is accomplished through an orbit virtual camera, providing different levels of details. Additionally, the data center experts can select a specific rack in order to explore information per server. The interactive elements of the scene include both the server racks and the server units. Upon hovering over a server rack (through mouse or gestures), a notification pops–up presenting basic information (i.e., the rack's id, the room it is located in, and its coordinates in the room, if the unit is in critical state). If the user selects a specific rack, the scene changes to a close–up view, the camera zooms–in towards the selected rack and specifically displays the unit of the rack which is in the most critical state. While in close-up view, the user may select any unit of a rack to view its properties. Furthermore, four different line charts are displayed for the selected unit, illustrating a history of the properties of the selected unit (temperature, CPU load, power consumption, network Bytes In/Out and Disk I/O).

Gesture-based Interaction. The application supports mouse–based and gesture-based interaction, aiming to address the potential contexts of use, i.e., a typical desktop environment and a control room. In order to make the interaction more natural and friendly to the user when the system is deployed in a large–screen setting, gesture-based interaction is supported featuring full hand articulation tracking [[10] and [11]]. Towards this goal, we developed a new approach for vision–based gesture recognition that encompasses hand detection, hand tracking, hand posture and hand gesture recognition. Aiming at intuitive means of interaction, we identified gestures consisting of both static and dynamic hand configurations that involve different scales of observation (from arms to fingers). The selection of the gestural vocabulary took into account the need for accurate gesture recognition regardless of the multifaceted variability of the human subjects performing them. Special attention was put so that the selected gestures are physically easy to be performed and intuitive enough to be remembered in daily routine. The current version implements a set of gestures through which the users can navigate through the 3D scene of the application, as well as hand-based mouse emulation for allowing the users to interact with specific UI controls.

When the user points towards the screen and moves his hand in the space, then a mouse cursor moves accordingly in the virtual 3D scene. In order to select a UI control, the user has to point with one hand to a particular component and then close and open the palm of the other hand in order to select it.

For navigation in the 3D space, the following gestures' vocabulary has been developed:

- Rotate Right/Left: The rotation to the sides involves steering an imaginative steering handle to one direction, i.e., pushing in front the right hand while pulling back the left hand in order to rotate left, and pushing in front the left hand while pulling back the right hand in order to rotate right. This gesture results into orbiting the camera of the virtual world in the corresponding direction.
- Rotate Up/Down: In order to rotate upwards or downwards, the user pushes with both palms open in an upwards or downwards direction, respectively. This gesture results into orbiting the camera of the virtual world in the corresponding direction.

- Zoom In/Out: For zooming-out of a view or exiting from the close-up view, the user pushes forward with both palms open, while for zooming–in they pull backwards with both palms closed.
- Increasing/Decreasing visualization density: The action of the user holding hands forward and opening or closing them results into increasing or decreasing how densely or sparsely the server racks are visualized in the 3D area.

3 Use Cases Validation

In this section, three use cases are described which validate LeanBigData and highlight the contribution of the platform's innovations and technological outcomes.

3.1 Cloud Data Centres: Data Centre Monitoring

Modern IT management systems employ a variety of models for monitoring and managing large IT infrastructures. These models range from relatively simple event–condition–action rules to sophisticated simulation and multivariate regression. Models may also represent the relationship between different elements in the system. These models reflect normal, or expected, behaviour of managed environment. They are employed for detecting anomalies or changes, when observed behaviour departs significantly from the expected behaviour given by a model, or for prediction in order to anticipate behaviour of the managed environment under changing load conditions and over time.

Often these models are created using domain knowledge of experts, however, with the popularization of ML techniques, the model creation process has more and more shifted to a process of model training using monitoring or experiment data collected over a period of time. Increasing sizes of data centres plus faster monitoring frequencies are creating large amounts of data that have to be stored and analyzed in an OLTP capable of handling a large data bandwidth.

Anomaly Detection and Root–cause Analysis. One of the objectives in the datacentre monitoring use case is to be able to identify events in the datacentre by looking at a small subset of the collected metrics (i.e., to perform Root-Cause Analysis). First of all, anomalies in these metrics, with respect to current accepted model for each server in the datacentre, have to be identified. Typically the output of such anomaly detectors is noticeably noisy. To produce a sensible root-cause analysis, the sequence of anomalies seen so far is considered by a Bayesian Network [12].

Since the components in a datacentre are frequently structured in a hierarchical way, the results of analysis in lower levels is aggregated on higher levels, which, by being able to observe the whole picture, can alter the low-level conclusions. In this way, the root cause analysis remains fast and accurate.

Results. The datacentre use case inputs data to the LeanBigData platform through its Complex Event Processing (CEP) subsystem. There the custom anomaly detection operators compare incoming data with predicted by the model values. When discrepancies are statistically significant, alerts are generated that go to the root-cause analysis operators, where the Bayesian inference takes place. In this process, both the input data stream as well as the alarms with their root-causes are stored in the SQL storage for later query and analysis by the human operators of the datacentre.

Visualization of results and current state of the datacentre is achieved using three different interfaces: a 2D eagle-view interface, a visual query builder for interactive query and chart visualization of results and a novel 3D HCI interface (see Sec. 2.7).

All in all, the LeanXcale platform provides the key ingredients (CEP capabilities, charts for query visualization, distributed relational storage, etc.) for the scalable datacentre management tool that is built in this use case.

3.2 Financial/Banking: Electronic Alignment of Direct Debit Transactions

Payment systems are rapidly evolving towards new technological solutions that bring as side effect new security vulnerabilities and weaknesses. As soon as a new payment method is adopted, fraudsters try to exploit security issues affecting it. European Union has developed the Single Euro Payments Area (SEPA), where 500 million of citizens, businesses and the European Public Administrations can make and receive over 100 billion no–cash payments every year. Regrettably SEPA Direct Debit (SDD) is vulnerable to cybercrime attacks, that have as preparatory action "Identity Theft", that can concern either the Debtor's identity or the Creditor's identity. A study conducted by Center of Economics and Business Research of Britain, showed that from 2006 to 2010 the Direct Debit frauds have increased of 288% [13]. According to a report published in 2015 by the European Central Bank card–not–present – e.g. direct debit and online payments – frauds are not only the largest category of frauds with 958 million of Euro losses, but also the only one recording an increase compared with the previous years [14]. The major weakness of the SEPA Direct Debit process is at the beginning of the procedure, specifically during the phase of signing the mandate. A fraudster can maliciously authorize the SDD mandate on behalf of the Debtor. In the last years, despite the recommendations from the European Banking Committee (EBC) to improve the security of SDD payment process, financial institutions have not implemented any effective solution to spot fraudulent transactions. In order to timely and effectively detect frauds in SDD transactions we propose a decision support system which is capable of gathering data provided by multiple sensors and correlating them by exploiting the Dempster-Shafer Theory (DS) [15]. The proposed architecture is shown in Figure 4 and includes the following functional blocks: Debtor Profile Creation, Online Transactions Categorization and Decision & Reaction.

Debtor Profile Creation – The first system component allows to build and store the profiles of debtors in a centralized database. The profile will be used to verify if the transaction characteristics meet the expected behaviour of the debtor. To create the profile, the system analyses data from several sources, such as debtor's social network accounts, customer banking records, questionnaires and third-party services. From the debtor's social network accounts (i.e. Facebook and Twitter) and through the combined use of text processing tools and a set of databases of well-known personalities, customer's interests will be determined and evaluated. For each interest (e.g. sports, technology, fashion, etc.) a belief and plausibility value will be set and associated to the debtor's

SCIENCE AND TECHNOLOGY PUBLICATIONS

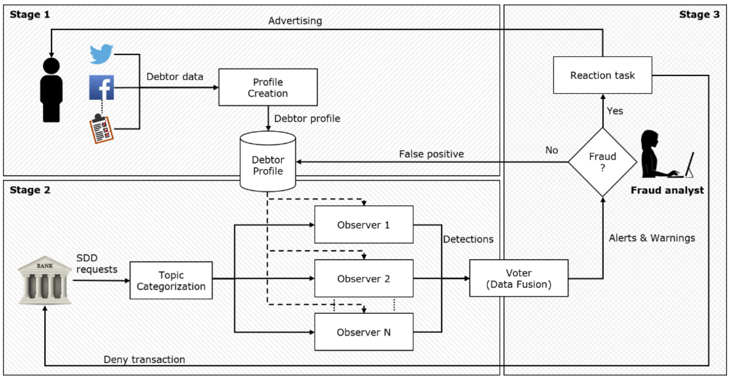

Fig. 4. Banking System Architecture.

profile. The current implementation of the component responsible for the debtor profile creation processes the following information: Facebook posts–likes–locations and Twitter hashtags–followers–tweets. Customer's banking records and a questionnaire could be periodically proposed by financial institutions to the account holders to refine their profiles. Online Transactions Categorization - This component is responsible for the assessment of SDD transactions and their classification as normal or anomalous events. To this end, a series of observers have been implemented. The topic categorization block in Figure 4 receives raw SDD requests, extracts the fields that identify the creditor and the reason for payment, processes this information, and assigns the transaction to one of topics/services specified by the bank. In case the block is not able to assign the transaction to any topic/service, a warning to fraud analyst is raised. The output of the topic categorization block will be correlated with the profile of the debtor involved in the transaction by using the following observers: Geographic Incoherence, Interests Incoherence, Creditor Reliability and Frequency Incoherence. The "Geographic Incoherence" is applicable to the transactions that involve the fruition of "location–bound" services ("location–bound" indicates the services that require the physical presence of the customer at the place where the services is used). This criterion measures the coherence between the debtor's addresses and the location where the service is provided. The "Interests Incoherence" observer can be used to detect suspicious SDD transactions both for "location–bound" and "location–unbound" services ("location–unbound" identifies services that can be provided at any location). It measures the deviation of the debtor's interests from the topic of the transaction being analysed. The "Creditor Reliability" observer allows to establish whether a company is real or not by using information provided by third-party services. The "Frequency Incoherence" observer relies on the fact that direct debit is a service typically used to perform recursive payments. The presence of spurious payments could be the symptom of suspicious transactions. Given the collected evidence and setting appropriate belief and plausibility thresholds, each observer provides mass probability values that can be interpreted as the results of the anomaly detection process applied to SDD transactions. Decision & Reaction - This block is responsible for making the final decision on whether a SDD transaction is a fraud or not. To do this the decision making process exploits data fusion and combines the output produced by the proposed observers (using DS rule of combination) in order to reduce the false–positive ratio and increase the performance of the detection activity.

Results. The proposed architecture has been implemented using the features provided by the LeanBigData platform: 1) Data Acquisition Framework: It is used to acquire and format raw data from Bank and from Social media, and to pre–process them; 2) CEP(Complex Event Processing) component: It is used by the Observers to perform the correlation between SDD data stream and Debtor/Creditor profiles. In particular, the queries that are loaded into the CEP have the following features: i) Real time query ii) Mixed query (Data Base + Data Stream); 3) Anomaly Detection component: It is used by the Voter to apply process mining techniques as well as subgraph patterns to perform anomaly detection. 4) Storage component (LeanXcale): It is used to store all information needed to detect the SDD frauds. 5) Visualization Component: It will be used to produce different graphical representations of data archived into the storage system, to provide a visual support to the identification of the SDD fraud.

3.3 Social Network Analytics

Social networks are becoming the most dynamic and easy-to-use mechanism to publish and exchange user–generated content. People use social networks for a variety of purposes, such as expressing opinions, spreading rumors, announcing gatherings, advertising products, etc. Among all social networks, Twitter has become the de facto source for real-time social media analytics due to a combination of factors: Twitter provides a platform for public short messages exchange used by billions of people, and a powerful open search and streaming APIs. Twitter is an invaluable data source for a variety of purposes, such as helping in real-time surveillance applications in leisure areas, brand monitoring in the retail sector or political trend observation during electoral campaigns.

Social Platform (LeanBigData Approach). Capturean [16] is a flexible, multipurpose, social network analysis solution developed by ATOS to take advantage of social networks to empower our client business. Some of its key features are:

- Fast and scalable data acquisition from (potentially) heterogeneous social networks.
- Execution of diverse analysis over the acquired data in order to generate several domain specific insights, both in streaming and batch fashion.
- Efficient as-a-service access to the data gathered and to the results of the data analysis.
- Flexible web-based GUI that allows easy extension / customization to meet distinct customers' needs.

A version of Capturean [17] has been adapted to run on top of LeanXCale, therefore improving its capabilities with LeanBigData framework features.

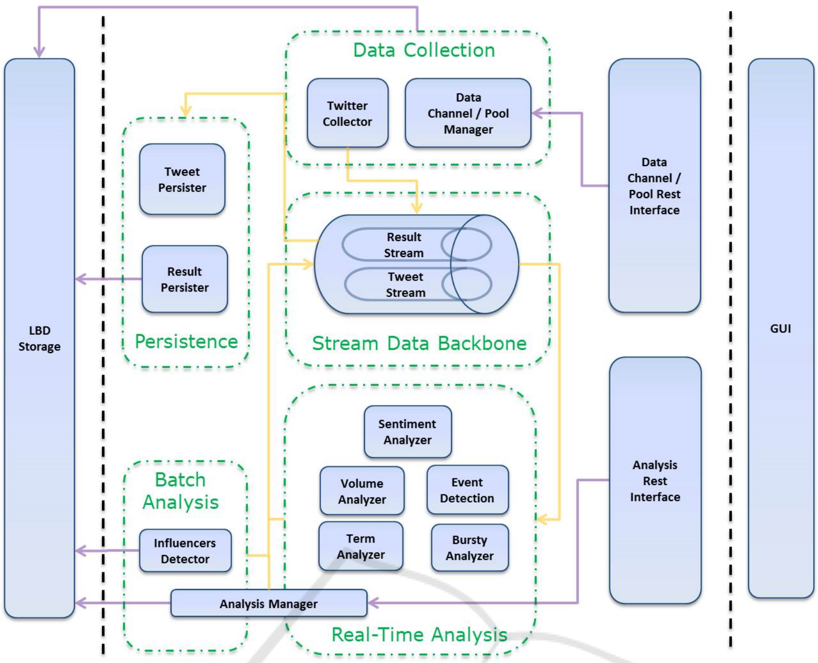

Fig. 5. Social Network Architecture.

Social Component. Capturean presents a modular architecture comprised of four main building blocks (data collection, real-time analysis, batch analysis and data persistence) connected by a central stream data backbone. The relevant building blocks showed in the figure 5 are:

SCIENCE AND TECHNOLOGY PUBLICATIONS

Data Collection. The Data Collection building block encompasses a set of modules that acquires data from a given social network and injects it into Capturean. Currently there are two components in this building block, allowing the data gathering from both Twitter Search and Stream APIs.
Real-time Analysis. Real-time analysis building block provides a set of algorithms used to process data streams coming from social networks. Several algorithms are available so far such us: term-occurrence analysis, volume analysis and sentiment analysis. In the scope of the project, two additional algorithms have been added to Capturean: sentiment analysis that will be described in later section, and bursty-word recognition, provided using LeanXCale anomaly detection module. Finally, Capturean provides an easy way to plug-in new stream analytics algorithms, therefore giving the possibility to extend the system with new analysis.
Batch Analysis. Batch analysis building block encompasses a set of algorithms used to process previously gathered data sets. The current implementation of Capturean includes influencer detection analysis and time based aggregation calculations. As in the case of real-time analysis, the system is easily extensible with new batch analytics.

Sentiment Analysis Component. LeanBigData Sentiment Analyzer is included in the real-time analysis building block providing real time sentiment analysis. The majority of the Social Networks show some inherent characteristics, such as sparsity, multilingualism, noise, neologisms, etc., that make the application of many language-processing methods not suitable for sentiment analysis. The algorithm developed in LeanBigData for Sentiment Analysis deals with the aforementioned issues by employing a variation of the n-gram graphs technique [18],supported by a voting ensemble and a manually annotated training set, aggregated from online surveys [19][20][21][22].Our training set was comprised of 4 million tweets concerning the Madrid local elections, retrieved during May 2015 using Twitter search API. This set was used to validate our Sentiment Analyzer performance, as well as our approach scalability when it comes to big loads of data.

The n-gram graphs approach was applied using the Weka data mining software [23] to train various classification algorithms (Naive Bayes Multinomial –NBM, Support Vector Machines – SVM, Logistic Regression –LReg, *C4.5* tree, Multilayer Perceptron – MLP and k-Nearest Neighbours). In general, such problems illustrate a trade-off between time performance and classification accuracy. The n-gram graphs method uses a low number of features (< 10), compared to usual methods, which results into a high classification time efficiency, a crucial requirement for streaming data, as in the case of the current use case scenario. Another technique we employed to improve efficiency is the discretization of classification features (n-gram graph similarities), which significantly improves component performance time-wise, and does not affect most algorithms' effectiveness.

All in all, *C4.5* and *LReg* were the ones that presented the best combination of effectiveness and efficiency among all the classification algorithms, thus these are the ones we chose to use in the LeanBigData Sentiment Analyzer.

SCIENCE AND TECHNOLOGY PUBLICATIONS

3.4 Targeted Advertisement

Advertisement Platform (LeanBigData Approach). Portugal Telecom(PT) sells multi-platform ads online covering the whole spectrum of web, mobile and TV. Similar to other industry standards, such as Google AdSense and AdWords, it allows advertisers to define their own campaigns, set their campaign goals and budget, their choice of paid words, as well as many other constraints including geographic and demographic of the targeted customers. Decisions on which ads to show in which client need to be made in a fraction of a second and should be informed by all the batch processing associated with profiling. To reliably provide efficient, contextualized and targeted advertisements to final users, both web services and application users, the current architecture of AdServer relies on in–house developed tools for handling the high-throughput stream of data and to deal with analysis and visualization. To cope with these large streams of data, PT currently uses a hodgepodge of big data technologies, leading to high communication overheads and undesired operational complexity. Additionally, this infrastructure processes about 90 million requests per day. Such requests can go up to 150 million requests per day in peak days, representing over 3600 requests per second. 90% of these requests need to be served in less than $30msec$. Ad Serving use–case is focused on serving targeted and contextualized ads to both clients and ad server actors. The major goals of this case study include: (i) simplification of the operational complexity; (ii) improvement of the overall efficiency and (iii) improvement of the throughput. A high-level overview of the integration of LeanBigData platform with the current infrastructure is depicted in Figure 6. This integration will allow a simplification of the mixture of technologies actually being used, as will allow significant improvements in both the throughput of data analysis as well as in the forecasting and profiling capabilities of the system. From the database perspective, the goal will be to take advantage of several improvements at the SQL-engine level [24] [25] as well as the new scalable fault-tolerant transactional platform specifically designed for OLTP workloads [26] to enhance the global throughput of the system. Additionally, recent work on a new approach for Key-Value Data Stores [27] will significantly improve OLAP workloads.

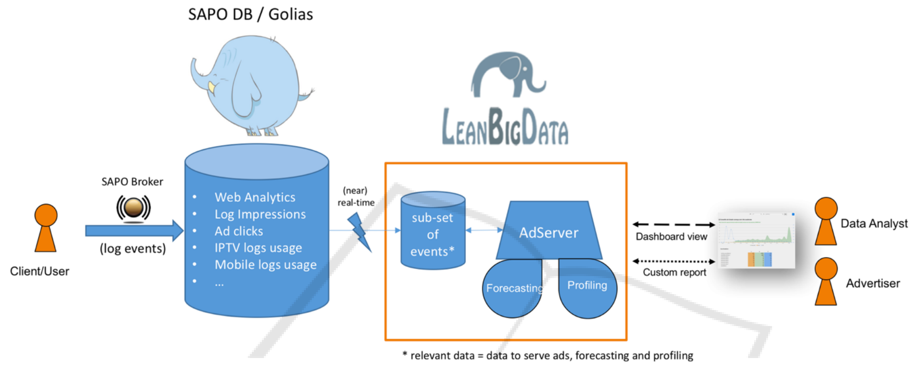

Fig. 6. Targeted Advertisement architecture.

The AdServer platform was emulated in a dedicated cluster of virtual machines, with the support of LeanBigData framework at an SQL boundary, and with a usage scenario as close as possible to the live system. The workload for the AdServer emulation consisted of 95% of impressions, 3% of dashboard queries and 2% of Forecasting. These indicators were extracted from the workload of the live AdServer, and they indicate that 95% of the requests are "inserts" queries with logging information from each impression. Second, dashboard OLAP queries embrace a significant 3% of the its workload and are used for batch and real-time dashboards. Finally, 2% of the workload is targeted to forecasting, typically integrated into dashboards, also represented by OLAP queries. For these benchmarks, three different scenarios were taken into account: (i) Production AdServer: live system currently being used at PT; (ii) Emulated AdServer: an emulated version of the Production AdServer in a dedicated cluster with similar workloads and hardware specification and (iii) LeanBigData AdServer: a testing version of PT AdServer use-case in LeanBigData platform. The conditions for the benchmarks are based on four different parameters: (i) number of users (200.000) represents the total number of different users that may request an ad; (ii) number of banners (100.000) represents the possible number of banners each request can deliver; (iii) number of interactions (100.000) concerns with the total number of impressions generated during the tests and (iv) number of connections, varying from 50 to 250, represents the number of concurrent requests to the AdServer.

Results. *Benchmark 1*: Emulated AdServer versus Production AdServer measures both the response time and throughput of the setups previously mentioned. For the first test the average response time vary – according to the number of concurrent clients – from $12.6ms$ to $19.4ms$, with a minimum of $5.4ms$, which is bellow the average response time for the Production AdServer, $13ms$. The throughput varies from 3668 to $2577 requests/sec$, with a maximum of $9259 requests/sec$. There results indicate that the Emulated AdServer achieved similar results compared against the Production AdServer.
Benchmark 2: Emulated AdServer versus LeanBigData AdServer is based on two distinct metrics: the throughput and response time. The throughput results achieved show that LeanBigData platform performed better that the Emulated AdServer from PT, with a different of approximately $1500 requests/sec$. This improvement is considerably interesting, since it represents almost 50% of the average throughput of events at Production AdServer ($3600 requests/sec$). For the response time metric results indicate that LeanBigData is still above the average response time for PT Emulated Adserver, with $5.4ms$. Nevertheless, ongoing work in LeanBigData project is expected to bring significant improvements in this matter, in particular with the integration of the new key-value data store [27] and other improvements in the SQL engine [26] [24].
Benchmark 3: LeanBigData AdServer scalability tests the scalability of LeanBigData platform, by progressively increasing the number of nodes from 1 to 5. The metrics for this benchmark are the average response time and the average throughput. The average response time obtained stabilizes when adding 2 more nodes to the AdServer configuration, indicating that this result is not affected by the increasing number of nodes and the potential overhead for management. For the throughput, results indicate that the system is capable of linearly increase its overall throughput capability with the increase of nodes in the AdServer configuration. Both tests are strong indicators that LeanBigData platform can in fact horizontally scale (by increasing the number of nodes) without affecting its performance.

4 Conclusions

The LeanBigData platform delivers a data management platform that provides real-time big data analytics capabilities by blending in a single database engine OLTP and OLAP capabilities. This blending is possible thanks to the ultra-scalability of the OLTP engine that can be scaled as much as needed to be able to run OLAP queries over the operational data. The OLAP engine has been attained by parallelizing the query plans and enabling their execution at multiple query engine instances. The OLAP engine implements both intra-query and intra-operator parallelism to be able to reduce the query processing time by distributing the load of large analytical queries across multiple query engine nodes. The platform is highly efficient thanks to the improvements made to the parallel-distributed transactional engine and the novel storage server technology, KiVi, developed in the project. The platform is integrated with a new distributed CEP engine that combined with the real-time analytics platform enables to address any data management problem based on both streaming data and data at rest. The LeanBigData platform comes with a visual monitor that enables to monitor the performance of the different layers of the platform and drill them down to resource utilization of individual nodes. The monitor leverages a sensor and data collector subsystem that is configurable and enables to adapt the data to be sent by sensors.

The LeanBigData platform includes a set of companion technologies to ease the development of analytical applications: visual analytics workbench and anomaly detection and root cause analysis. The visual analytics workbench enables to do explorative business analytics by transforming the result set of analytical queries and visualizing it. The anomaly detection and root cause analysis enable to detect exceptional situations and find out the reason for them to perform automatic tasks in different contexts (e.g. detecting a fraud or an issue in a data centre). The LeanBigData platform has been validated by four different uses cases data centre monitoring, eAdvertisement, fraud detection in direct debit operations and social network sentiment analysis.

Acknowledgments. This work has been supported by the LeanBigData[4] FP7 European project (grant agreement no 619606) and the CoherentPaaS[5] FP7 European project (grant agreement no 611068),and partially funded by the Regional Government of Madrid, project Cloud4BigData (S2013/ICE-2894) co-funded by FSE & FEDER and by the Spanish Research Council (MICCIN), project BigDataPaaS (TIN2013-46883)

References

1. Ahmad, Y., Berg, B., Çetintemel, U., Humphrey, M., Hwang, J., Jhingran, A., Maskey, A., Papaemmanouil, O., Rasin, A., Tatbul, N., Xing, W., Xing, Y., Zdonik, S.B.: Distributed operation in the borealis stream processing engine. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, June 14-16,. (2005) 882–884
2. Pu, C., Schwan, K., Walpole, J.: Infosphere project: System support for information flow applications. SIGMOD Record 30(1) (2001) 25–34
3. Gulisano, V., Jimenez-Peris, R., Patino-Martinez, M., et al.: Streamcloud: An elastic and scalable data streaming system. IEEE Trans. Parallel Distrib. Syst. 23(12) (2012) 2351–2365
4. Rana, A.I., Estrada, G., Solé, M.: Big data stream clustering algorithms empirical evaluation. In: Final Public Workshop from LeanBigData and CoherentPaaS, RTPBD, DiscoTec, Greece. (2016)
5. Ruiz, C., Menasalvas, E., Spiliopoulou, M.: C-denstream: Using domain knowledge on a data stream. In Gama, J., Costa, V., Jorge, A., Brazdil, P., eds.: Discovery Science. Volume 5808 of Lecture Notes in Computer Science. Springer Berlin Heidelberg (2009) 287–301
6. Aggarwal, C.C., Han, J., Wang, J., Yu, P.S.: A framework for clustering evolving data streams. In: Proceedings of the 29th International Conference on Very Large Data Bases - Volume 29. VLDB '03, VLDB Endowment (2003) 81–92
7. Drossis, G., Birliraki, C., Patsiouras, N., Margetis, G., Stephanidis, C.: 3d vizualization of large scale data centres. In: Proceedings of the 6th International Conference on Cloud Computing and Services Science. (2016) 388–395
8. Drossis, G., Grammenos, D., Birliraki, C., Stephanidis, C. In: MAGIC: Developing a Multimedia Gallery Supporting mid-Air Gesture-Based Interaction and Control. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 303–307

4 http://leanbigdata.eu
5 http://coherentpaas.eu

SCIENCE AND TECHNOLOGY PUBLICATIONS

9. Drossis, G., Grammenos, D., Adami, I., Stephanidis, C. In: 3D Visualization and Multimodal Interaction with Temporal Information Using Timelines. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 214–231
10. Kyriazis, N., Oikonomidis, I., Panteleris, P., Michel, D., Qammaz, A., Makris, A., Tzevanidis, K., Douvantzis, P., Roditakis, K., Argyros, A. In: A Generative Approach to Tracking Hands and Their Interaction with Objects. Springer International Publishing, Cham (2016) 19–28
11. Rautaray, S.S., Agrawal, A.: Vision based hand gesture recognition for human computer interaction: a survey. Artificial Intelligence Review 43(1) (2015) 1–54
12. Pearl, J.: Probabilistic reasoning in intelligent systems - networks of plausible inference. Morgan Kaufmann series in representation and reasoning. Morgan Kaufmann (1989)
13. Direct debit fraud at an all time high. https://www.finextra.com/newsarticle/22028/direct-debit-fraud-at-an-all-time-high-bacs-challenges-figures [Online; accessed 19-July-2016].
14. European Central Bank: Fourth report on card fraud. https://www.ecb.europa.eu/pub/pdf/other/4th_card_fraud_report.en.pdf [Online; accessed 19-July-2016].
15. Dempster, A.P.: A generalization of bayesian inference. Journal of the Royal Statistical Society. Series B (Methodological) (1968) 205–247
16. Capture Software. http://capturean.atosresearch.eu/ [Online; accessed 19-July-2016].
17. Flink Software. https://flink.apache.org/ [Online; accessed 19-July-2016].
18. Aisopos, F., Tzannetos, D., Violos, J., Varvarigou, T.: Using n-gram graphs for sentiment analysis: An extended study on twitter. In: 2016 IEEE Second International Conference on Big Data Computing Service and Applications (BigDataService), IEEE (2016) 44–51
19. Shamma, D.A., Kennedy, L., Churchill, E.F.: Tweet the debates: understanding community annotation of uncollected sources. In: Proceedings of the first SIGMM workshop on Social media, ACM (2009) 3–10
20. Narr, S., Hulfenhaus, M., Albayrak, S.: Language-independent twitter sentiment analysis. Knowledge Discovery and Machine Learning (KDML), LWA (2012) 12–14
21. Tromp, E.: Multilingual sentiment analysis on social media. Lap Lambert Academic Publ (2012)
22. Villena Román, J., Lana Serrano, S., Martínez Cámara, E., González Cristóbal, J.C.: Tass-workshop on sentiment analysis at sepln. (2013)
23. Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., Witten, I.H.: The weka data mining software: an update. ACM SIGKDD explorations newsletter 11(1) (2009) 10–18
24. Coelho, F., Pereira, J.O., Vilaça, R., Oliveira, R.: Holistic shuffler for the parallel processing of sql window functions. In: Distributed Applications and Interoperable Systems - 16th IFIP WG 6.1 International Conference, DAIS 2016, Held as Part of the 11th International Federated Conference on Distributed Computing Techniques, DisCoTec 2016, Heraklion, Crete, Greece, June. (2016) 75–81 ¡p¿n/a¡/p¿.
25. Gonçalves, R.C., Pereira, J., Jiménez-Peris, R. In: An RDMA Middleware for Asynchronous Multi-stage Shuffling in Analytical Processing. Springer International Publishing, Cham (2016) 61–74
26. Jiménez-Peris, R., Patiño-Martínez, M., Kemme, B., Brondino, I., Pereira, J.O., Vilaça, R., Cruz, F., Oliveira, R., Ahmad, M.Y.: Cumulonimbo: A cloud scalable multi-tier SQL database. IEEE Data Eng. Bull. 38(1) (2015) 73–83
27. Ahmad, M.Y., Kemme, B., Brondino, I., Patiño-Martínez, M., Jiménez-Peris, R. In: Transactional Failure Recovery for a Distributed Key-Value Store. Springer Berlin Heidelberg, Berlin, Heidelberg (2013) 267–286

SCIENCE AND TECHNOLOGY PUBLICATIONS